A Value Equivalence Approach for Solving Interactive Dynamic Influence Diagrams

Ross Conroy[*], Yifeng Zeng[*], Marc Cavazza[**], Jing Tang[*], Yinghui Pan[***]
[*]Teesside University, Middlesbrough, UK
{ross.conroy, y.zeng, x9019186}@tees.ac.uk
[**]University of Kent, Kent, UK
m.o.cavazza@kent.ac.uk
[***]Jiangxi University of Finance and Economics, China
panyinghui@jxufe.edu.cn

ABSTRACT

Interactive dynamic influence diagrams (I-DIDs) are recognized graphical models for sequential multiagent decision making under uncertainty. They represent the problem of how a subject agent acts in a common setting shared with other agents who may act in sophisticated ways. The difficulty in solving I-DIDs is mainly due to an exponentially growing space of candidate models ascribed to other agents over time. in order to minimize the model space, the previous I-DID techniques prune behaviorally equivalent models. In this paper, we challenge the minimal set of models and propose a value equivalence approach to further compress the model space. The new method reduces the space by additionally pruning behaviorally distinct models that result in the same expected value of the subject agent's optimal policy. To achieve this, we propose to learn the value from available data particularly in practical applications of real-time strategy games. We demonstrate the performance of the new technique in two problem domains.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

Influence Diagrams, Decision Making, Multiple Agents

1. INTRODUCTION

Interactive dynamic influence diagrams (I-DIDs) [7, 24] provide a general framework for solving sequential multiagent decision making problems under uncertainty. Different from other frameworks, like Dec-POMDPs [19] and multiagent influence diagrams (MAIDs) [10], I-DIDs solve the problem from the perspective of individual agents and do not make a common belief assumption on modeling other agents. Hence I-DIDs become a more general decision model and can be employed to solve both cooperative and competitive multiagent decision problems. Recent research has found some practical applications of I-DIDs [12, 11, 4].

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

Algorithms for solving I-DIDs need to compute a large number of candidate models of other agents that represent how the agents optimize their decisions in an uncertain environment. In addition, the I-DID solutions have to track the evolution of all the models as other agents observe, act and update their beliefs over time. Consequently, the computational complexity of solving I-DIDs is mainly due to the exponential growth in the number of models are ascribed to other agents.

Recognizing that only the predicted behavior of other agents matters to the decisions of the subject agents, existing I-DID solutions [24] reduce model space by retaining only those models that exhibit different behavior of other agents. For example, the exact algorithm, called discriminative model update (DMU) [6], clusters models that are *behaviorally equivalent* (BE) - whose behavioral predictions for the other agent are identical - and maintains one representative model from every cluster. Exploiting BE models has become a core technique for reducing the complexity of I-DID solutions. Continuous effort has focused on identifying BE models and resulted in a series of approximate I-DID algorithms [26]. Recently, Chen *et al.* [3] develop an online algorithm that limits the search in the model space by incrementally building the true models of other agents during the agents' interactions. Conroy *et al.* [4] learn behavior of other agents from interaction data and don't need to explicitly model decision making process for other agents. This line of work improves the BE-based I-DID solutions and facilitates the I-DID applications in practice. However, all the previous techniques still rest on the BE principle for reducing the model space of other agents. As a large amount of different behavior always exists for agents, the existing I-DID algorithms are not sufficiently capable to deal with complex decision making problems.

In this paper, we initiate a new approach on compressing model space of other agents in I-DIDs. The behavioral equivalence focuses only on the difference of other agents' behavior and postulates that the difference leads to distinction of the subject agent's decisions. However, it is still safe to assume that the difference of other agents' behavior may only matter if it affects the expected value of the subject agent's optimal policy for their interactions. In other words, what really matters to the subject agent is not the difference of other agents' behavior, but the difference of the received rewards. Inspired by this observation, we redefine the equivalence of other agents' models in terms of their impact on the subject agent's expected value. Models that generate the same expected value of the subject agent's optimal policy are value equivalence (VE) and can be grouped into one VE class. The new VE measurement in comparison to BE will further reduce the model space since distinct behavior of other agents may generate the same

expected values to the subject agent and will be clustered. In addition, VE has a direct link with the solution quality, which has been lacking in the previous I-DID techniques.

Without building a complete I-DID, it is not easy to compute expected values of the subject agent. We resort to agents' interaction history and learn the expected values for determining VE of models ascribed to other agents over time. This is particularly useful for applications where interaction data is widely available and will be incrementally added to over time. As demonstrated in recent I-DID applications in computer games [4], the continuously uploaded data that records gaming activities of computer-and-human players facilitates the learning task in the new VE approach. In this context, this paper makes the following contributions:

- We propose a new approach for solving I-DIDs. The VE technique reduces the model space of other agents by considering their behavioral impact on expected values of the subject agent. This provides more reduction than the previous I-DID solutions.
- We focus on learning expected values from agents' interaction data and develop approximate techniques for determining the model equivalence.
- We theoretically analyze computational savings of the new model compression technique compared to BE. Additionally we demonstrate the performance in a set of experiments and focus on practical applications of computer games.

We organize the paper as follows. We briefly review the I-DID framework as well as the BE concept in Section 2. We formulate the new approach of value equivalence and propose a learning technique for determining VE in Section 3. The computational savings and solution quality are theoretically analyzed in Section 4. We empirically analyze the method performance in a scalable UAV (unmanned aerial vehicle reconnaissance problem) simulation testbed and demonstrate applications in a real-time strategy game in Section 5. We discuss related work in Section 6 and conclude this paper with a discussion of the challenges and future work.

2. BACKGROUND: INTERACTIVE DYNAMIC INFLUENCE DIAGRAM

We start with a brief review on the framework of interactive dynamic influence diagrams (I-DIDs) with elaboration in the well-studied multiagent tiger problem [8]. Subsequently, we describe the I-DID solutions based on behavioral equivalence (BE).

2.1 Representation

I-DIDs represent sequential decision making problems for a subject agent who interacts with other agents in an uncertain environment. As other agents act simultaneously, which is not fully observable to the subject agent, I-DID models their predicted behavior by solving all possible models of other agents. Actions of both the agents impact the common environmental states S and rewards R. Fig. 1 shows a level l I-DID for the subject agent i who models other agent j in level $l-1$, where level refers to recursive reasoning between agents and agents in level 0 do not model the others. In addition to regular chance, decision and utility nodes in DID [20], a new type of node called the *model node*, $M_{j,l-1}$, models how other agent j makes its decisions simultaneously at level $l-1$. More explicitly, it contains a set of j's candidate models whose solutions give the predicted behavior A_j, which is represented by a *policy link* (the dashed line) connecting $M_{j,l-1}$ and A_j. Each candidate model of agent j, $m_{j,l-1}$, could be either a level $l-1$ I-DID or a DID at level 0.

The I-DID modeling complexity arises with update of the model node (containing j's models) over time, as represented by the *model*

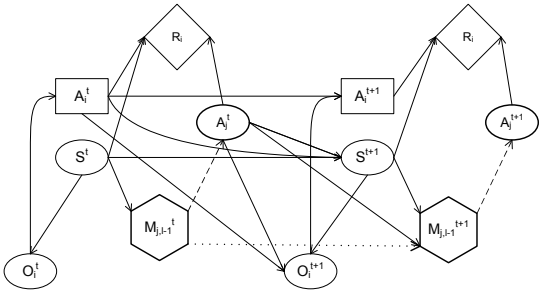


Figure 1: A generic two time-slice level l I-DID for agent i who optimises its decisions A_i given observations O_i.

update link (the dotted arrow from $M^t_{j,l-1}$ to $M^{t+1}_{j,l-1}$ in Fig. 1). As agent j acts and receives observations over time, the models are updated to reflect its changed beliefs. The updated models differ in the beliefs of j's actions and observations. Since agent i needs to track the update of j's models, the number of models grows in a new model node. The number of models in $M^{t+1}_{j,l-1}$ is up to $|\mathcal{M}^t_{j,l-1}||A_j||\Omega_j|$ where $|\mathcal{M}^t_{j,l-1}|$ is the number of models at time step t, and $|A_j|$ and $|\Omega_j|$ are the largest spaces of actions and observations respectively.

We may replace the model nodes and model update links with regular chance nodes and dependency links in I-DID. Subsequently, I-DID becomes a regular DID and any DID technique can be used to solve the converted I-DID. Below we use a two-agent tiger problem to elaborate the I-DID framework.

Figure 2 shows a level 1 I-DID for agent i who considers two models of agent j, $m^{t,1}_{j,0}$ and $m^{t,2}_{j,0}$, at level 0. The converted I-DID is a regular DID in which the chance node $Mod[M_{j,0}]$ represents agent j's possible models. The models differ in j's beliefs about the tiger's location and solving the models obtains optimal decisions for agent j. As indicated by the conditional probability table (CPT) in Fig. 4, agent j's optimal decisions are OL and L respectively when the two models are solved at level 0. The optimal decisions are mapped into the predicted actions in the chance node A^t_j.

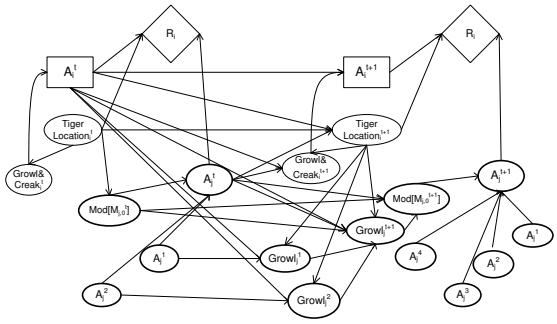


Figure 2: A converted level 1 I-DID for agent i in the tiger problem.

We show the update of $m^{t,1}_{j,0}$ and $m^{t,2}_{j,0}$ in Fig. 3. As agent j may receive one of two observations (either GL or GR), four new models are generated in the model node $M^{t+1}_{j,0}$.

We show the CPT of $Mod[M^{t+1}_{j,0}]$ in Fig. 4. For example, the first row of the CPT shows that $m^{t,1}_{j,0}$ is updated into the model $m^{t+1,1}_{j,0}$ when agent j takes the action OL at time t and observes GL at $t+1$. As neither OR nor L is the optimal decision for $m^{t,1}_{j,0}$, we assign a uniform distribution to indicate that $m^{t,1}_{j,0}$ does not transform into any of the new models for these actions.

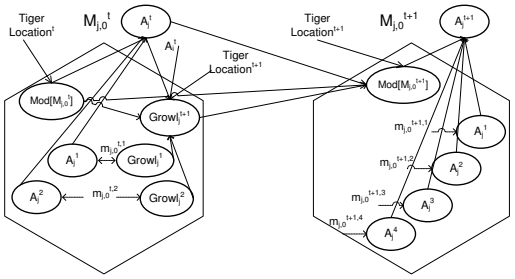


Figure 3: Details of the model update link where two models are expanded into four models in $M_{j,0}^{t+1}$.

Mod[$M_{j,0}^t$]	OL	OR	L
$m_{j,0}^{t,1}$	1	0	0
$m_{j,0}^{t,2}$	0	0	1

CPT of A_j^t

Decisions(A_j)
OL: Open the left door
OR: Open the right door
L: Listen
Observations(Growl$_j$)
GL: Growl from the left door
GR: Growl from the right door

<A_j^t, Growl$_j^{t+1}$>	Mod[$M_{j,0}^t$]	$m_{j,0}^{t+1,1}$	$m_{j,0}^{t+1,2}$	$m_{j,0}^{t+1,3}$	$m_{j,0}^{t+1,4}$
<OL, GL>	$m_{j,0}^{t,1}$	1	0	0	0
<OL, GR>	$m_{j,0}^{t,1}$	0	1	0	0
<L, GL>	$m_{j,0}^{t,2}$	0	0	1	0
<L, GR>	$m_{j,0}^{t,2}$	0	0	0	1
<OR, * >	*	1/4	1/4	1/4	1/4
<L, * >	$m_{j,0}^{t,1}$	1/4	1/4	1/4	1/4
<OL, * >	$m_{j,0}^{t,2}$	1/4	1/4	1/4	1/4

CPT of Mod[$M_{j,0}^{t+1}$]

Figure 4: The CPTs of the chance nodes A_j^t and $Mod[M_{j,0}^{t+1}]$.

2.2 Solutions and Behavioral Equivalence

We present the exact I-DID algorithm in Fig. 5. We first transform an I-DID into a regular DID by expanding j's models at level $l-1$ (lines 2-15) and then solve the converted DID (lines 16-18). Lines 4-5 solve j's models to instantiate the policy link. Line 6 invokes techniques for compression of the model space based on *behavioral equivalence* [18], **PruneBehavioralEq** ($\mathcal{M}_{j,l-1}$), and returns representative models of j. Lines 7-15 implement the model update link in the I-DID. Finally, lines 17-18 solve the transformed I-DID using standard DID algorithms.

Previous I-DID techniques focus on implementing BE in either exact or approximate ways. Formally we define behavioral equivalence of agent j's models below.

DEFINITION 1 (BEHAVIORAL EQUIVALENCE). *Two models, m_j and $\hat{m}_j$, of agent j, are behaviorally equivalent if* $\mathsf{OPT}(m_j) = \mathsf{OPT}(\hat{m}_j)$, *where* $\mathsf{OPT}(\cdot)$ *denotes the solution of the model.*

A model solution is the agent's policy and is generally represented by a policy tree. A depth-T policy tree contains a set of policy paths, $\mathcal{T}_j^T = \bigcup h_j^T$ where the policy path, h_j^T, is an action-observation sequence over T planning horizons. We let $h_j^T = \{a_j^t, o_j^{t+1}\}_{t=0}^{T-1}$, where o_j^T is null with no observations following the final action. Notice that a policy tree can either be built by solving an agent's model or be learned from the available data that describes the agent's behavior [4].

Thus, BE models are those whose behavioral predictions for agent j are identical. After compressing the BE models, the procedure **PruneBehavioralEq** ($\mathcal{M}_{j,l-1}$) returns a set of representative models that are behaviorally distinct. The set of behaviorally distinct models, denoted by $\hat{\mathcal{M}}_{j,l-1}^{BE}$, are considered as the minimal set of agent j's behavior [24]. In this paper, we aim to further compress the model space by merging behaviorally distinct models.

I-DID EXACT(level $l \geq 1$ I-DID or level 0 DID, horizon T)
Expansion Phase
1. **For** t **from** 0 **to** $T-1$ **do**
2. **If** $l \geq 1$ **then**
Populate $M_{j,l-1}^{t+1}$
For each m_j^t **in** $\mathcal{M}_{j,l-1}^t$ **do**
4. Recursively call algorithm with the $l-1$ I-DID (or DID) that represents m_j^t and horizon, $T-t$
5. Map the decision node of the solved I-DID (or DID), $OPT(m_j^t)$, to the corresponding chance node A_j
6. $\mathcal{M}_{j,l-1}^t \leftarrow$ **PruneBehavioralEq**($\mathcal{M}_{j,l-1}^t$)
7. **For each** m_j^t **in** $\mathcal{M}_{j,l-1}^t$ **do**
8. **For each** a_j **in** $OPT(m_j^t)$ **do**
9. **For each** o_j **in** O_j (part of m_j^t) **do**
10. Update j's belief, $b_j^{t+1} \leftarrow SE(b_j^t, a_j, o_j)$
11. $m_j^{t+1} \leftarrow$ New I-DID (or DID) with b_j^{t+1}
12. $\mathcal{M}_{j,l-1}^{t+1} \overset{\cup}{\leftarrow} \{m_j^{t+1}\}$
13. Add the model node, $M_{j,l-1}^{t+1}$, and the model update link
14. Add the chance, decision, and utility nodes for $t+1$ time slice and the dependency links between them
15. Establish the conditional probability tables (CPTs) for each chance and utility node
Solution Phase
16. **If** $l \geq 1$ **then**
17. Represent the model nodes, policy links and the model update links to obtain the DID
18. Apply the standard look-ahead and backup method to solve the expanded DID

Figure 5: Algorithm for exactly solving a level $l \geq 1$ I-DID or level 0 DID expanded over T time steps.

3. VALUE EQUIVALENCE APPROACH AND IMPLEMENTATION

Due to the uncertainty of agent j's models, I-DID algorithms are challenged by the exponential growth in the number of the models over time. BE literally compares solutions of j's models and maintains only the behaviorally distinct models that may become a sufficient coverage of j's behavior.

As different behavior of agent j may have the same impact on the subject agent's decisions, grouping behaviorally distinct models may further reduce the model space without compromising the I-DID solution quality. We utilize this insight toward developing a new technique on examining the model equivalence in the I-DIDs.

3.1 Value Equivalence

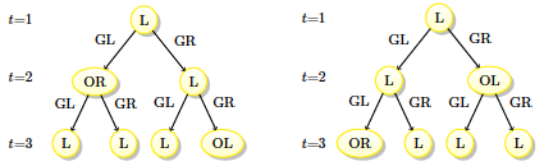
We assume that models of agent j have identical frames and differ only in their beliefs in I-DID. Our aim is to identify models that are *value equivalence* (VE) from the perspective of the subject agent i.

The expected value of level l agent i's optimal policy given by the I-DID for T time steps is computed in Eq. 1.

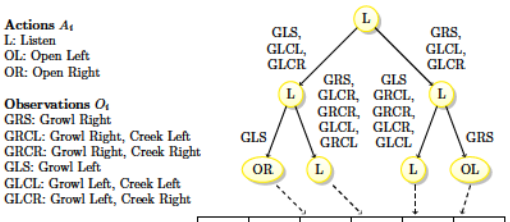
$$V^T(m_{i,l}) = \rho(b_{i,l}, a_i^*) + \sum_{o_i} Pr(o_i|b_{i,l}, a_i^*) V^{T-1}(m'_{i,l}) \quad (1)$$

where $\rho(b_{i,l}, a_i^*) = \sum_{s, m_{j,l-1}} b_{i,l}(s, m_{j,l-1}) \sum_{a_j} R_i(s, a_i^*, a_j) \times Pr(a_j|m_{j,l-1})$. Here, $b_{i,l}(s, m_{j,l-1})$ is the agent i's belief over the physical states and possible models of j at level $l-1$, a_i^* is i's optimal action and $m'_{i,l}$ is the updated model of agent i containing the updated belief at the next time step.

Let $m_{j,l-1}, \hat{m}_{j,l-1} \in \mathcal{M}_{j,l-1}$ be two candidate models of agent j at level $l-1$, and $\bar{\mathcal{M}}_{j,l-1}$ be the set of j's candidate models



(a) Agent j's policy tree by solving m_j^0 (b) Agent j's policy tree by solving m_j^1



m_j^0	Pr(·)	0.20225	0.30255	0.3026	0.1926
-	EV(·)	1.351	-0.8682	-0.8691	1.695
m_j^1	Pr(·)	0.19455	0.30335	0.3032	0.1989
-	EV(·)	1.494	-0.798	-0.798	1.469

(c) Agent i's policy given by the I-DID

Figure 6: Agent i has the same expected value of the optimal policy (c) for different j's models: m_j^0 (a) and m_j^1 (b).

excluding $m_{j,l-1}$ and $\hat{m}_{j,l-1}$,
e.g. $\bar{\mathcal{M}}_{j,l-1} = \mathcal{M}_{j,l-1}/(m_{j,l-1}, \hat{m}_{j,l-1})$. The two models are value equivalence if enclosing either of them in the I-DID will result in the same expected value for agent i. Let $V^T(m_{i,l}|m_{j,l-1})$ (or $V^T(m_{i,l}|\hat{m}_{j,l-1})$) be i's expected value when the model $m_{j,l-1}$ (or $\hat{m}_{j,l-1}$), together with other models $\bar{\mathcal{M}}_{j,l-1}$, are included in the I-DID and are subsequently expanded over T time steps. Formally we define the value equivalence (VE) of two models below.

DEFINITION 2 (VALUE EQUIVALENCE). *Two models of agent j, $m_{j,l-1}$ and $\hat{m}_{j,l-1}$, are value equivalence if $V^T(m_{i,l}|m_{j,l-1}) = V^T(m_{i,l}|\hat{m}_{j,l-1})$, where $V^T(m_{i,l}|m_{j,l-1})$ and $V^T(m_{i,l}|\hat{m}_{j,l-1})$ are as defined above.*

In other words, VE models are those that induce identical expected values for agent i when the I-DID is expanded using any of the models. Thus, the VE models do not enforce the behavioral equivalence of the models. We elaborate it below.

EXAMPLE 1. *Assume that three j's models are expanded in an I-DID of three time steps in the two-agent tiger problem. Fig. 6(a) and (b) show the policy trees for two models with different initial beliefs. The models are not behaviorally equivalent since they have different actions except those at t=1. As shown in the table, agent j's behaviors from different models induce different probabilities ($Pr(h_i)$) over agent i's action-observation sequences every one of which is a policy branch in the tree. However, each of which generate the same expected value of agent i's optimal policy given by the I-DID, which is the sum of the expected values for all policy paths ($EV(h_i)$). Thus, the two models are VE. For the case of I-DIDs with 5 planning horizons, using VE can reduce more than half of behaviorally distinct models of agent j.*

Given the set of agent j's candidate models, $\mathcal{M}_{j,l-1}$, we can group the models that are confirmed to be VE, and then pick a representative model while pruning others from the group. The representative models from the different groups are value difference and partition the entire model space of agent j. Immediately we may introduce an approximate version of VE in Definition 3. A larger value of ϵ groups more value difference models thereby resulting in less model space of agent j.

DEFINITION 3 (ϵ-VE). *Two models of agent j, $m_{j,l-1}$ and $\hat{m}_{j,l-1}$, are ϵ-VE if $|V^T(m_{i,l}|m_{j,l-1}) - V^T(m_{i,l}|\hat{m}_{j,l-1})| \leqslant \epsilon$.*

Similarly to the BE approaches, the procedure **PruneBehavioralEq** $(\mathcal{M}_{j,l-1})$ can be replaced with the one using VE to prune the models in the model node. Recall that agent i assigns some probability mass to each model in the model node ($Mod[M_j^t]$). When the VE models are pruned, we transfer over from the probability mass over the pruned models to the representative models that are retained in the model node. This avoids introducing errors in the I-DID solution quality due to the loss of probability mass over j's candidate models.

3.2 Value Computation

To determine VE of agent j's models, we need to compute the expected value of agent i's optimal policy given by the I-DID that will be constructed with the expansion of all j's models. This seems to be a paradox since we need to first prune the VE models and then expand the I-DID accordingly. A potential method that interleaves VE determination with a partial I-DID expansion could be developed, which will be discussed in Section 7.

In this paper, we focus on learning VE from available domain data. This is partially motivated by the I-DID practical applications in a real-time strategy game (RTS) where game replay data is continuously supplied by the growing gamer community [4]. Without explicitly building decision making models for agents, we can learn their policies/behaviors from the data. We compute VE from input data instead of candidate models, because computing VE from candidate models would require all such models to be solved as part of the I-DID expansion. I-DID expansion can be time consuming especially for larger time horizons, sometimes impossible due to computational and memory limitations. Computing VE from data avoids this computational complexity allowing for a reduced candidate set of models to be created without I-DID expansion. We will present the VE learning technique in the computer game context, which is also applicable in the setting where agents' interaction data is available.

3.2.1 Value of the Learned Policy

In the context of RTS games, we build an I-DID from the perspective of a non-player character (NPC, or one human-player, denoted by agent i) that models human-players (agent j) at a low level. The NPC aims to optimize its policy upon predicting behavior of human-players. Since it is rather difficult to explicitly model the decision making process of human-players through specific decision models, like IDs or DIDs, we learn their behavior from replay data. Each type of behavior is corresponding to one possible model of human-players. By doing this, we don't need to build the descriptive models of human-players and then solve the models to obtain their behavior. As different types of human-players exist in the game community, the learned behaviors could be many and the I-DID for the NPC cannot include all types of the behavior. We will reduce the behavioral space directly through the VE approach. In a popular RTS game, namely *StarCraft* [1], we elaborate the computation of the expected value of agent i in the VE approach.

Table 1 shows a portion of replay data publicly available in *StarCraft*. The data records the game time-stamp, unit identifier and type, units' rewards in the current game state and so on. The units

[1] http://eu.blizzard.com/en-gb/games/sc/

Time	Unit ID	Type	...	Obs	Action	Util
1	216	Goliath	...	3	escape	0
1	212	Goliath	...	3	escape	0
			...			
5	216	Goliath	...	1	attack	1

Table 1: Sample of *StarCraft* replay data recording activities of various units during the gameplay.

are controlled by either the NPC or human-players. As presented in [4], the behavior of each unit over T time steps can be built as a policy tree in which the probability of each policy path is calculated accordingly. The probability is computed as the occurring frequency of an action-observation sequence in the interaction activities.

Let $Pr(h_i^T)$ be the probability of a policy path in the tree, and $U(h_i^T)$ be the rewards that are gathered by the agent executing the action-observation sequence. We can compute $U(h_i^T)$ by summing immediate rewards received by the agent over the entire planning horizon. Subsequently, we can calculate the expected value of agent i's policy as follows.

$$V^T(\mathcal{T}_i) = \sum_{h_i^T \in \mathcal{T}_i} Pr(h_i^T) U(h_i^T) \tag{2}$$

Given sufficient interaction data, the learned behavior becomes the optimal policy of agents. The expected value of the learned behavior can be counted as the expected value as it is computed by solving its corresponding model, e.g., $V^T(m_{i,l}) = V^T(\mathcal{T}_i)$. Without differentiating whether the policy is obtained by either solving the agent's models or learning the behavior from the data, we denote the expected value as $V^T(m_{i,l} \sim \mathcal{T}_i)$.

As shown in [4], we can still learn the agent's policy from limited data that is sufficiently good to be used in the I-DIDs. With the increasing set of interaction data, the learned policy approaches the optimal policy for agents. The quality of the I-DID solutions can be bounded with a probabilistic guarantee, which may in turn ensure the quality of the VE determination.

3.2.2 Implementation

We build a level l I-DID, $m_{i,l}$, for an NPC (agent i) where the variables (including states, observations, actions and rewards) are retrieved from the data and follow gaming knowledge. Since the set of variables are obtained by following a unique type of unit, we can learn the agent i's policy, $\mathcal{T}_i$, from the replay data. As the policy is interleaving with various types of human-players (agent j), we compute its expected value for each type of j's behavior, e.g., $V^T(m_{i,l} \sim \mathcal{T}_i | \mathcal{T}_{j,l-1})$. Consequently, we can obtain a set of the expected values, $\{V^T(m_{i,l} \sim \mathcal{T}_i | \mathcal{T}^1_{j,l-1}), \cdots, V^T(m_{i,l} \sim \mathcal{T}_i | \mathcal{T}^n_{j,l-1})\}$, each of which quantifies the impact of one type of human-players' behavior on the NPC's policy. By comparing the expected values, we can identify VE of human-players' behavior and prune the behavioral space. The reduced set of agent j's behavior is used to expand the I-DID $m_{i,l}$, as developed in the expansion phase of the I-DID algorithm in Fig. 5.

4. SAVINGS AND SOLUTION QUALITY

As with the previous I-DID techniques, the primary complexity of solving I-DIDs is due to the exponentially growing number of j's models over time. At time step t, there could be $|\mathcal{M}^0_{j,l-1}|(|A_j||\Omega_j|)^t$ many models of the other agent j, where $|\mathcal{M}^0_{j,l-1}|$ is the number of models considered initially. The previous BE methods have reduced the model set into the minimal set, $\hat{\mathcal{M}}^{BE}_{j,l-1}$, that is much smaller than $\mathcal{M}^0_{j,l-1}$. Since VE further clusters behaviorally distinct models, it results in less model space. Let $\hat{\mathcal{M}}^{VE}_{j,l-1}$ be the largest set of value difference models. Then, the following proposition holds.

PROPOSITION 1 (CARDINALITY). *The set $\hat{\mathcal{M}}^{VE}_{j,l-1}$ resulting from the VE approach is not larger than the behaviorally distinct set $\hat{\mathcal{M}}^{BE}_{j,l-1}$ from the BE approach.*

PROOF. BE requires that models have the same optimal actions given observations for each time step. As indicated in the value computation in Eq. 2, the BE models will result in the same expected value. Thus, the BE models naturally become the VE models. Meanwhile, as discussed previously, the behavioral difference could be mediated by the receiving rewards of agent i in the value computation. Thus, the behaviroally distinct models may result in the same expected value and are further classified as VE models. The VE approach may filter out more models than BE does. □

The BE techniques fail to measure the quality of I-DID solutions, which are agent i's policy given by the I-DID, since BE literally compares j's policies that have no direct links with i's policy. In contrast, the VE approach conducts the model reduction by comparing the model influence on the expected value of i's policy. Hence the method can directly bound solution errors if approximation is introduced in the VE determination. Let $\hat{V}^T(m_{i,l})$ be expected value of i's optimal policy given by the I-DID in which the model $m_{j,l-1}$ is replaced with the representative VE model. Evidently, according to Def. 3, the ϵ-VE approach bounds the I-DID solution error not larger than ϵ. Thus, Proposition 2 holds.

PROPOSITION 2 (QUALITY). $|V^T(m_{i,l}) - \hat{V}^T(m_{i,l})| \leqslant \epsilon$

5. EXPERIMENTAL RESULTS

We implemented the VE approach by learning the expected values from the data. The implementation replaces the procedure **PruneBehavioralEq** ($\mathcal{M}_{j,l-1}$) in Fig. 5 to prune agent j's models in the I-DID. Meanwhile, we implemented the ϵ-VE method for the comparison purpose. We compare the variants of the VE approach to the exact BE technique (DMU) [6] and report their performance in two problem domains. The first domain is the multi-unmanned aerial vehicle reconnaissance problem (UAV), which is the largest problem setting so far used in the recent I-DID development [24]; while the second one is the RTS game of *StarCraft* in the I-DID practical applications. We show that, (a) in comparison to DMU, the VE approach further reduces the model space of other agents in the I-DID and achieves better scalability; (b) the quality of solutions provided by the VE methods improves upon more available data; (c) the VE technique can be effectively adapted in an online I-DID solution and outperforms the recently developed online I-DID solution [3].

5.1 UAV Problem Domain

We assume that the UAV scenario is played out in a 5×5 grid of sectors ($|S|$=81, $|A_i|$=$|A_j|$=5, $|\Omega_i|$=$|\Omega_j|$=5), as illustrated in Fig. 7. We build the level 1 I-DID for UAV I modelling J using level 0 DIDs. We consider 20 models of UAV J that differ in the beliefs on its initial position in the grid.

We solve the I-DID using the DMU approach and obtain the optimal policy for UAV I. To generate the data for the VE approach, we let UAV I play with J for N times in which a model of J is randomly picked in the interaction. Given the data of N plays, we

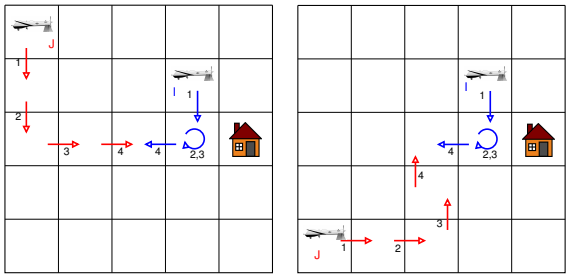


Figure 7: UAV I is tasked with intercepting J before the latter raids the allied base. Two example trajectories show the different behavior of UAV J, which results in the same interception strategy for UAV I. The numbers denote the policy steps of the two UAVs that move continuously and concurrently in the simulation.

proceed to learn J's behavior as well as I's policy, and use VE to prune the behavioral space. Subsequently, we build a new I-DID using the J's behavior reduced by the VE approach. Solving the I-DID results in the new I's optimal policy.

In Fig. $8(a)$-(b), we show the average rewards received by UAV I over 200 runs when it plays with J by executing the policies obtained by either DMU or VE approaches. Since DMU can't solve the I-DIDs for the horizon of 7 [2], its performance is not shown in Fig. $8(b)$. When more interaction data is available for learning the values, determining VE becomes more accurate and the performance approaches that of the DMU method. The ϵ-VE methods perform as expected for the solution quality when the ϵ values are varied in the experiments. We notice that the VE approach even outperforms DMU given more accumulated data. This is because the DMU approach considers the entire space of UAV J's distinct behavior with a uniform probability distribution while VE concentrates on a small set of J's behavior that avoids more randomness in I's prediction.

Fig. $8(c)$ confirms our intuition that VE leads to fewer model classes compared to DMU. A larger ϵ value allows the grouping of more VE models resulting in more compressed model space at each time step. We show two examples of selected trajectories of UAVs I and J using the two methods for a horizon of 4 in Fig. 7. Although UAV J exhibits rather different strategies on approaching the allied base, the distinction does not lead to a different policy to I intercepting J before J initiates the attack. Note that the two example trajectories represent some typical raid behavior of UAV J, which belongs to solutions of J's different models differing in its initial beliefs.

Since VE leads to a smaller model space, it achieves a better scalability than DMU and can solve the I-DID for a larger horizon up to 10 [3]. We don't report their time efficiency in solving the I-DIDs since DMU and VE employ different schemes to obtain J's policies. But we observe in the experiments that learning behavior from data is much more efficient than solving decision models ascribed to J. Solving the I-DIDs that are expanded with the reduced number of models for other agents is very efficient.

We take a further step to adapt VE in the online I-DID solutions, namely OPIAM (online plan, interact and adapt models), as developed in [3]. OPIAM starts with a small set of J's models and uses BE to adapt the model space in the online interactions. In the version of OnlineVE, we adapt the model space by choosing the models that result in larger expected values to I during the interactions. In this set of experiments, we let UAV I choose a set of 5 models (from 20 models) of J to build an initial I-DID, and adapt the I-DID online. Fig. 9 shows that the OnlineVE progresses much better than OPIAM over interactions. OnlineVE benefits from directing I's prediction on J's behavior to I's rewards during immediate interactions. We are optimistic that VE may be well integrated into the development of other I-DID solutions.

5.2 StarCraft Application

The real world domain we choose to model and test the VE approach is *StarCraft*. We choose *StarCraft* because it has partial observability as to the true battle state, as well as the availability of human vs human replay files from sources such as Gosu Gamers [4] and Team Liquid [5]. From these replay files states, observations, actions and rewards can be extracted such as shown in Table 1 from which we can learn policies of human players.

StarCraft games are incredibly complex with players having to focus on areas such as resource gathering, build orders, combat and scouting. To simplify this domain for the purposes of this paper we focus on a typical combat scenario between groups of units. Specifically for testing we use a 3 vs 3 unit scenario. Fig $10(a)$ shows the complexity of games between two human players and the typical scenario we have broken this down into. We mine for data from replays by observing in the data where small groups of units are close together and record their states, actions, observations and rewards for these battles. A battle is considered over when either one of the groups of units are all killed or units have moved far enough away from enemy units to no longer engage in battle.

We build level 1 I-DID for player i modelling j of sectors ($|S|$=16, $|A_i|$=$|A_j|$=3, $|\Omega_i|$=$|\Omega_j|$=4) using data from replay files. Policies of player j are learned from replay data and pruned through either DMU or VE approaches. In Fig. $10(b)$ we show a small sample tree of typical behavior in the aforementioned scenario taken from a much larger and complex policy. The behavior shown is a typical defensive behavior where a player may be defending an area or unit by only attacking when confident there is a low number of enemy units to prevent loss, standing ground both when there is nothing close-by to attack, and when there is a larger number close-by standing ground to defend whatever the units may be defending.

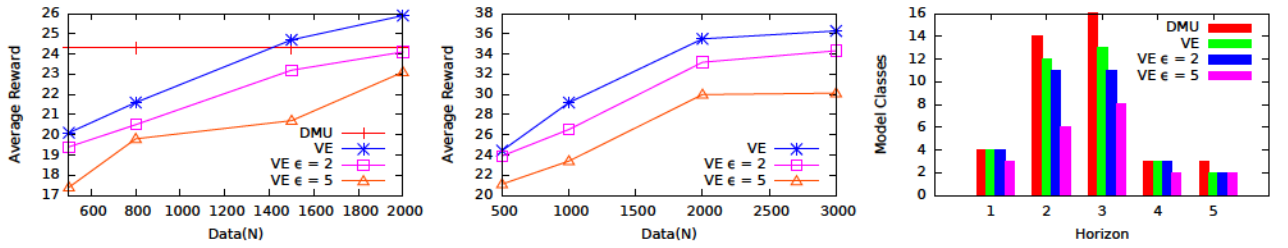
Fig. 11 shows results of experiments of simulated battles between players i and j where i is controlled by the polices by solving the above I-DIDs for varying amounts of mined replay data and j executes a random policy from those learned from the mined data, average rewards are calculated over approximately 100 battles. In Fig. $11(a)$-(b), we compare DMU methods of reducing j's model space with reduction of j's model space by VE and VE with varying ϵ values (0.25 and 0.3) for planning horizons T=5 and T=7. For the larger horizon of T=7 where more data is available a greater number of j's behavior are learned increasing the size of the I-DID model space. This causes the model to no longer be solvable by DMU methods due to memory limitations; however, we find that reduction by VE methods reduces the model space by approximately 20-30%, enough for the I-DID to be solved where DMU fails. We also find that for a larger ϵ such as 0.3 for VE methods can introduce unpredictability into the quality of policies calculated for i due to over-reducing the known behavior of j preventing accurate predictions of their behavior. On the other hand, VE performs better and more stable when more data is supplied.

[2] We use the approximate BE technique (ϵ-BE [24]) to generate the data.

[3] We use ϵ-BE to generate a sufficiently large set of data that result in reliable policies.

[4] http://www.gosugamers.net/

[5] http://www.teamliquid.net/



(a) Average reward for T=5 (b) Average reward for T=7 (c) Number of models at each time step for T=5

Figure 8: Performance profile for the UAV problems obtained by solving level 1 I-DID through either DMU or VE approaches.

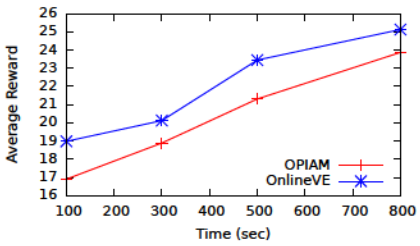


Figure 9: Average rewards received by UAV I interacting with J online. UAV I adapts the model space every 200-300 seconds.

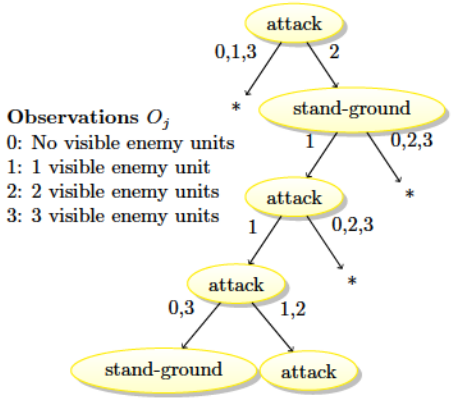
6. RELATED WORK

I-DIDs generalize influence diagrams to multiagent settings facilitating decision making in the presence of other sophisticated decision makers of uncertain types. They are viewed as graphical counterparts of finitely-nested interactive partially observable Markov decision processes (I-POMDPs) [8].

As we mentioned before, a predominant factor in the complexity of I-DIDs is due to an exponential growth in candidate models of other agents. Exploiting BE to reduce the model space is a mainstream research on addressing I-DID solution challenges [24]. Notably, by discriminating between model updates, the DMU approach [6] generates a minimal set of models in each non-initial model node. It may pre-emptively avoid expanding models that will turn out to be BE to others in the next time step. Meanwhile, much effort has been invested into determining BE models efficiently by investigating the development of policy trees. Zeng *et al.* [25, 5, 23] sought to cluster models by comparing only a partial set of paths in the policy trees. Chen *et al.* [3] initiated the study of online I-DID solutions by developing true behavior of other agents during their interactions. Conroy *et al.* [4] focused on learning agents' behavior from available data, which provides prior knowledge on refining model space in I-DIDs. The BE based techniques have improved the usability of I-DIDs and driven potential real-world applications [12, 11, 4], which contributes into the learning techniques in computer games [2].

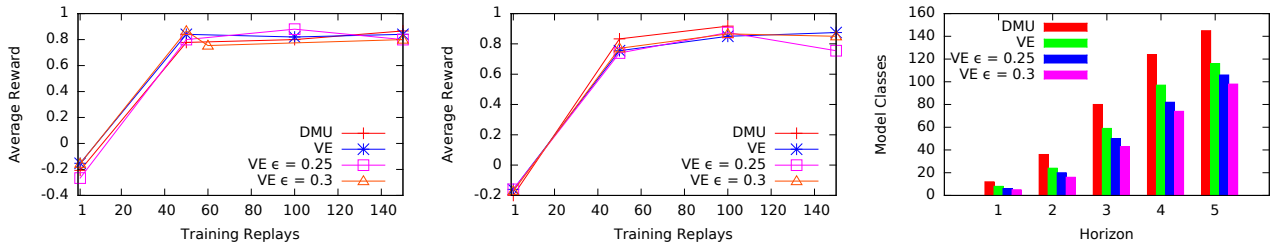
While graphical models remain as yet unexplored in the context of cooperative decision making models using frameworks such as decentralized POMDPs [19], factored representations of the state space are becoming prevalent [13]. The factored representations facilitate solutions to decentralized POMDPs with many agents by exploiting the interaction structure among the agents [14]. Pajarinen and Peltonen [16] utilized factored representations in a dynamic Bayesian network to project agents' beliefs forward, and applied expectation-maximization to learn stochastic finite-state controllers. Meanwhile, Witwicki and Durfee [22] used influence-

(a) *StarCraft* battles are very complex and we focus on a commonly occurring scenario where two groups of units are battling in the field.



(b) Sample section of a policy tree learned for agent j for a typical type of gaming behavior.

Figure 10: Applications of I-DIDs in *StarCraft* where behavior of players can be learned from replay data.



(a) Average reward for T=5 (b) Average reward for T=7 (c) Model reduction for T=5 from 150 replays

Figure 11: Performance profile of the VE approach in *StarCraft* where we learn behavior of player j from replay data.

based abstraction to decouple local agents' interactions in decentralized POMDPs, which is further generalized to quantify the complexity of multiagent planning [15].

Specifically, exploiting structure of value functions has been a useful technique for improving solutions to agents' planning [21]. This research attempts to reduce the complexity of belief updates by approximating the value functions, e.g., through a set of basis functions [9]. With the exploitation of stochastic transition and observation functions, the value function approximation results in very efficient solutions to decentralized POMDPs with many agents [17]. Recently, Amato and Oliehoek [1] used search methods to facilitate the decomposition of interacted values in multiagent planning. The VE technique shows significant improvement on solution scalability and enjoys theoretical guarantee.

7. DISCUSSION AND CONCLUSION

We show how we exploit behavioral impact to further determine the model equivalence and apply it to scale up solutions of I-DIDs. Our insight is that comparing the expected value induced by other agents is likely sufficient for grouping more models than previously clustered by BE approaches. We formulate a principled VE technique to decide the model equivalence and implement it through the value learning task. We empirically examine the VE approach from multiple facets in two selected problem domains.

While we demonstrate the utility of VE in a comprehensive set of experiments, we still face the challenge of computing the expected value of the subject agent without fully expanding the I-DID. As indicated in the experiments, learning values from data can serve to compose an initial model space that will be refined in the new interactions. This could be particularly useful in the areas of game playing and user modeling where either data or domain knowledge can be accessed. Pynadath and Marsella [18] demonstrated an application of utility equivalence techniques in a social simulation setting related to class bullying. Here, both the teacher and the bully maintain a limited number of mental models of each other without suffering a loss in expected utility.

Interleaving VE determination with expanding I-DIDs is another way to solve I-DID in a more general decision making setting. We evaluate VE while expanding the I-DID with a partial set of candidate models, and prune the models in an incremental way. The issue is about selection of candidates so that the solution quality could be guaranteed in the VE approach. This is in line with our future research.

REFERENCES

[1] C. Amato and F. A. Oliehoek. Scalable planning and learning for multiagent POMDPs: Extended version. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, pages 1995–2002, 2015.

[2] K. T. Andersen, Y. Zeng, D. D. Christensen, and D. Tran. Experiments with online reinforcement learning in real-time strategy games. *Applied Artificial Intelligence: An International Journal*, 23:855–871, 2009.

[3] Y. Chen, P. Doshi, and Y. Zeng. Iterative online planning in multiagent settings with limited model spaces and pac guarantees. In *Proceedings of the Fourteenth Internationl Conference on Autonomous Agents and Multiagents Systems Conference (AAMAS)*, pages 1161–1169, 2015.

[4] R. Conroy, Y. Zeng, M. Cavazza, and Y. Chen. Learning behaviors in agents systems with interactive dynamic influence diagrams. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 39–45, 2015.

[5] P. Doshi, M. Chandrasekaran, and Y. Zeng. Epsilon-subject equivalence of models for interactive dynamic influence diagrams. In *WIC/ACM/IEEE Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2010.

[6] P. Doshi and Y. Zeng. Improved approximation of interactive dynamic influence diagrams using discriminative model updates. In *Eighth Internationl Conference on Autonomous Agents and Multiagents Systems Conference (AAMAS)*, pages 907–914, 2009.

[7] P. Doshi, Y. Zeng, and Q. Chen. Graphical models for interactive pomdps: Representations and solutions. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 18(3):376–416, 2009.

[8] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research (JAIR)*, 24:49–79, 2005.

[9] M. Hauskrecht. Value-function approximations for partially observable markov decision processes. *Journal of Artificial Intelligence Research (JAIR)*, 13:33–94, 2000.

[10] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1027–1034, 2001.

[11] J. Luo, H. Yin, B. Li, and C. Wu. Path planning for automated guided vehicles system via interactive dynamic influence diagrams with communication. In *9th IEEE International Conference on Control and Automation (ICCA)*, pages 755 –759, 2011.

[12] B. Ng, C. Meyers, K. Boakye, and J. Nitao. Towards applying interactive POMDPs to real-world adversary modeling. In *Innovative Applications in Artificial Intelligence (IAAI)*, pages 1814–1820, 2010.

[13] F. Oliehoek, M. Spaan, S. Whiteson, and N. Vlassis.

Exploiting locality of interaction in factored Dec-POMDPs. In *Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 517–524, 2008.

[14] F. A. Oliehoek, S. Whiteson, and M. T. Spaan. Approximate solutions for factored Dec-POMDPs with many agents. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems (AAMAS)*, pages 563–570, 2013.

[15] F. A. Oliehoek, S. J. Witwicki, and L. P. Kaelbling. Influence-based abstraction for multiagent systems. In *Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, pages 1422–1428, 2012.

[16] J. Pajarinen and J. Peltonen. Efficient planning for factored infinite-horizon Dec-POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 325–331, 2011.

[17] J. Pajarinen and J. Peltonen. Efficient planning for factored infinite-horizon Dec-POMDPs. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence (IJCAI)*, pages 325–331, 2011.

[18] D. Pynadath and S. Marsella. Minimal mental models. In *Twenty-Second Conference on Artificial Intelligence (AAAI)*, pages 1038–1044, Vancouver, Canada, 2007.

[19] S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *Journal of Autonomous Agents and Multi-agent Systems*, pages 190–250, 2008.

[20] J. A. Tatman and R. D. Shachter. Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):365–379, 1990.

[21] T. Veiga, M. T. J. Spaan, and P. U. Lima. Point-based pomdp solving with factored value function approximation. In *Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, pages 2513–2519, 2014.

[22] S. J. Witwicki and E. H. Durfee. Influence-based policy abstraction for weakly-coupled dec-pomdps. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 185–192, 2010.

[23] Y. Zeng, Y. Chen, and P. Doshi. Approximating behavioral equivalence of models using top-k policy paths (extended abstract). In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1229–1230, 2011.

[24] Y. Zeng and P. Doshi. Exploiting model equivalences for solving interactive dynamic influence diagrams. *Journal of Artificial Intelligence Research (JAIR)*, 43:211–255, 2012.

[25] Y. Zeng, P. Doshi, Y. Pan, H. Mao, M. Chandrasekaran, and J. Luo. Utilizing partial policies for identifying equivalence of behavioral models. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 1083–1088, 2011.

[26] Y. Zeng, H. Mao, Y. Pan, and J. Luo. Improved use of partial policies for identifying behavioral equivalence. In *Proceedings of the Eleventh Internationl Conference on Autonomous Agents and Multiagents Systems Conference (AAMAS)*, pages 1015–1022, 2012.